CLAIMS

We claim:

1. A method comprising:

executing an exchange function comprising changing a local pointer for a global pointer;

jumping to an address contained in the local pointer wherein, the local pointer points to the exchange function, or the local pointer points to an address accessing a shared resource; and

executing the exchange function after accessing the shared resource.

2. The method of claim 1 wherein the exchange function is a method of a class including a property comprising the global pointer.

3. A computer system comprising plural multi-tasking processes performing the method of claim 1.

4. A computer system comprising plural time-sharing threads performing the method of claim 1.

5. The method of claim 1 wherein the exchange function is executed in executive mode.

6. The method of claim 1 wherein the local pointer is associated with a process.

7. The method of claim 1 wherein the local pointer is associated with a thread.

8. The method of claim 1 wherein the local pointer is a property of a class associated with a thread.

9. The method of claim 1 wherein the exchange function is a method of a class including a property comprising the local pointer, and the class is in a threads address space.

10. The method of claim 1 wherein the exchange function is a method of a class including a property comprising the global pointer, and the class is in a processes address space.

11. Plural child threads of a process in contention for the shared resource, and performing the method of claim 1.

12. The method of claim 11 wherein child threads each include a local variable, and the process includes the global variable.

13. The method of claim 1 wherein the exchange function is executed as an atomic unit.

14. A computer-readable medium comprising instructions for performing the method of claim 1.

15. A computer-readable medium including instructions for performing functions comprising:

creating a global variable initialized with a shared resource access address;

creating threads including a local variable initialized with an exchange function address;

the exchange function, for exchanging a threads local variable with the global variable;

control flow for transferring control to an address contained in a threads local variable; and

control flow for calling the exchange function after accessing the shared resource.

16. The computer-readable medium of claim 15 wherein the function for creating the global variable further comprises creating as an atomic unit.

17. The computer-readable medium of claim 15 wherein the shared resource is a memory location.

18. The computer-readable medium of claim 15 wherein the exchange function further comprises calling and returning from a sleep function before exchanging variables.

19. The computer-readable medium of claim 15 wherein during execution of the method by a created thread, control flow for transferring control to the address contained in the thread's local variable, transfers control to one of the exchange function or the shared resource.

20. A computer system comprising:

a processor coupled to memory;

plural units of execution sharing the processor and performing a method comprising, executing an exchange function comprising changing a local pointer for a global pointer;

jumping to an address contained in the local pointer wherein, the local pointer points to the exchange function, or the local pointer points to an address accessing a shared resource; and

executing the exchange function after accessing the shared resource.

21. The computer system of claim 20 wherein said plural units of execution are threads, and a first thread after executing the exchange function, jumps to the address accessing the shared resource, and before the first thread exits the shared resource, control of the processor transfers to a second thread, and the second thread upon executing the exchange function, jumps to the address of and cycles in the exchange function.

22. The computer system of claim 21 wherein after control transfers to the first thread, and after the first thread exits the shared resource and executes the exchange function, and after control of the processor transfers to the second thread, the second thread executes the exchange function and jumps to the address accessing the shared resource.

23. The computer system of claim 20 wherein the exchange function is a method of a class including a property comprising the global pointer.

24. The computer system of claim 20 wherein the exchange function is executed in user mode.

25. The computer system of claim 20 wherein the units of execution are processes.